

FINAL PROJECT SEMESTER GANJIL 2022/2023
MATA KULIAH SISTEM BASIS DATA (SBD)
KELAS A KELOMPOK 1



Nama anggota:

X	X
X	X
X	X
X	X
X	X

Teknik *fact-finding*:

Observe, (unstructured) interview, dan research

Organisasi tujuan:

Yayasan Kanker Indonesia (YKI) Cabang Koordinator Jawa Timur

Narasumber *interview*:

Khairun Sani, S.Psi. (Kepala Kesekretariatan)

DEPARTEMEN SISTEM INFORMASI
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA

DAFTAR ISI

DAFTAR ISI.....	2
INSTRUMEN.....	3
What facts are collected?.....	3
Unstructured Interview.....	4
Observe.....	5
Research.....	5
REQUIREMENTS.....	6
Umum.....	6
Kamar.....	7
Pekerja: Pengurus dan Volunteer.....	7
Donatur.....	9
Inventaris.....	9
DATABASE DESIGN PROCESS.....	10
Enhanced Entity-Relationship (EER) Model.....	10
Relational Schema.....	11
Conceptual Design.....	12
Logical Design.....	12
Physical Design.....	13
Database Management System (DBMS).....	13
DDL (Data Definition Language).....	13
DML(Data Manipulation Language).....	20
DQL (Data Query Language).....	24

INSTRUMEN

- Menyajikan gambaran umum studi kasus Yayasan Kanker Indonesia Jawa Timur (YKI) dan kemudian menggunakan studi kasus ini untuk mengilustrasikan cara mendirikan proyek basis data.
- Mengilustrasikan bagaimana teknik Fact Finding dapat digunakan dan dokumentasi yang dihasilkan pada tahap awal siklus pengembangan sistem basis data:
 - Perencanaan basis data,
 - Definisi sistem, dan
 - Tahap pengumpulan dan analisis kebutuhan

What facts are collected?

Tahap Database SDLC	Contoh data yang diambil	Contoh dokumentasi yang dihasilkan
Perancangan basis	Maksud dan tujuan basis data	Pernyataan misi dan proyek
Definisi sistem	Deskripsi tampilan pengguna utama (termasuk peran pekerjaan atau area lamaran bisnis)	Definisi ruang lingkup dan batas aplikasi basis data; Definisi tampilan pengguna yang akan didukung
Pengumpulan dan analisis persyaratan	Persyaratan untuk tampilan pengguna; spesifikasi sistem, termasuk persyaratan kinerja dan keamanan	Spesifikasi persyaratan pengguna dan sistem
Desain basis data	Tanggapan pengguna untuk memeriksa desain basis data logis; fungsionalitas yang disediakan oleh DBMS target	Desain basis data konseptual / logis (termasuk model ER, kamus data, dan skema relasional); Desain Database Fisik
Desain Aplikasi	Tanggapan pengguna untuk memeriksa desain antarmuka	Desain aplikasi (termasuk deskripsi program dan antarmuka pengguna)
Seleksi DBMS	Fungsionalitas yang	Evaluasi dan

	disediakan oleh respons Pengguna DBMS target terhadap prototipe	rekomendasi DBMS
Prototyping	Tanggapan pengguna terhadap prototipe	Persyaratan pengguna dan spesifikasi sistem yang dimodifikasi
Implementasi	Fungsionalitas yang disediakan oleh DBMS target	
Konversi dan pemuatan data	Format data saat ini; Kemampuan impor data dari target DBMS	Konversi dan pemuatan data
Pengujian	Hasil tes	Desain aplikasi (termasuk deskripsi program dan antarmuka pengguna)
Pemeliharaan operasional	Hasil pengujian kinerja; Persyaratan pengguna dan sistem baru atau yang berubah	Panduan pengguna; analisis hasil kinerja; Persyaratan pengguna yang dimodifikasi dan spesifikasi sistem

Unstructured Interview

Teknik *unstructured interview* digunakan untuk mengumpulkan keperluan. Pertanyaan-pertanyaan pada *interview* disampaikan dengan bahasa mengalir yang tidak terlalu teknis dan dapat dimengerti oleh orang awam. Konsep-konsep yang mendasari pertanyaan-pertanyaan yang ditanyakan adalah sebagai berikut.

- Jika tersedia, dapat diikuti dengan *observe* formulir dan lingkungan YKI
 - Dari formulir dapat dilihat data apa saja yang dikumpulkan.
- Diarahkan ke *key*, *simple*, *composite*, *multivalued*, dan spesifikasi lainnya
 - Apakah suatu atribut unik (dan dijadikan sebagai patokan data)?
 - Apakah suatu atribut memiliki *value* lebih dari satu?
 - Apakah suatu atribut memiliki bagian lebih kecil (*composite*)?
- Diarahkan ke *relationships* antarentitas
 - Apakah suatu atribut terhubung dengan entitas lainnya?
 - Apakah jenis hubungan data tersebut? *One-to-one*? *One-to-many*? *Many-to-many*?

Kemungkinan arah wawancara (hanya format/struktur data, bukan isi data):

- Data pasien rumah singgah
- Data rumah singgah

- Data kepengurusan
- Data keuangan

Rekaman wawancara (9 Oktober 2023) dapat diakses melalui:

Observe

Untuk mendapatkan fakta lapangan sebagai *requirements* dan bahan pertanyaan lanjutan untuk *interview* maka juga digunakan teknik *observe*. Instrumen yang diobservasi berupa formulir dan media lain yang digunakan untuk menghimpun data. Media lain yang dimaksud mencakup papan pencatatan, *file* Microsoft Excel, dan lain-lain.

Research

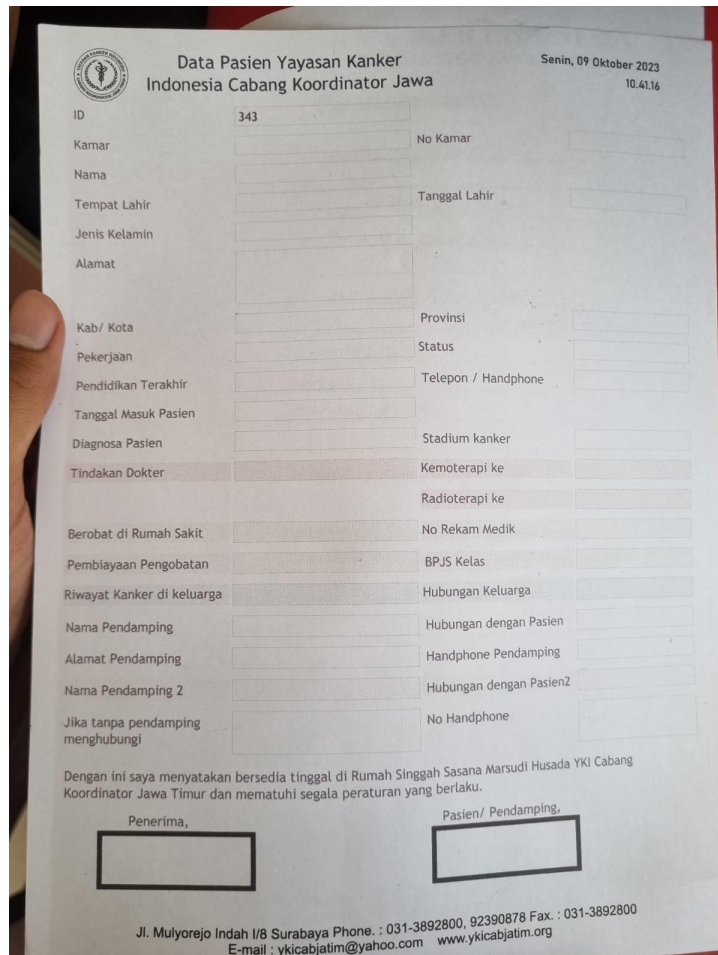
Dilakukan *research* melalui situs web www.ykijawatimur.org untuk mendapatkan data sebagai berikut.

Nama Data	URL
Data Penghuni Rumah Singgah 2022	https://www.ykijawatimur.org/data-penghuni-rumah-singgah-2022/data-penghuni-rumah-singgah-2022
Data Penghuni Rumah Singgah 2023	https://www.ykijawatimur.org/data-penghuni-rumah-singgah-2022/data-penghuni-rumah-singgah-2023 .
Ketentuan Donasi	https://www.ykijawatimur.org/donasi
Susunan Pengurus	https://www.ykijawatimur.org/tentang-yki/susunan-pengurus

REQUIREMENTS

Umum

- Yayasan Kanker Indonesia (YKI) mengkhususkan diri dalam upaya penanggulangan kanker, dengan mengupayakan penanggulangan kanker melalui berbagai kegiatan di bidang promotif, preventif, dan suportif. Hal tersebut didasarkan atas kepedulian dan keprihatinan terhadap semakin banyaknya penderita kanker, serta tingginya angka kematian penderita akibat berobat pada stadium lanjut.
- Pada tahun 2022, YKI Jawa Timur memiliki sekitar 183 pasien baru yang didaftarkan ke dalam 4 tipe kamar. Ketika seorang pasien datang ke kantor, formulir pendaftaran pasien digunakan. Formulir pendaftaran pasien untuk YKI ditampilkan dalam Gambar 1.



**Data Pasien Yayasan Kanker
Indonesia Cabang Koordinator Jawa**

Senin, 09 Oktober 2023
10.41.16

ID	343	
Kamar		No Kamar
Nama		
Tempat Lahir		Tanggal Lahir
Jenis Kelamin		
Alamat		
Kab/ Kota		Provinsi
Pekerjaan		Status
Pendidikan Terakhir		Telepon / Handphone
Tanggal Masuk Pasien		
Diagnosa Pasien		Stadium kanker
Tindakan Dokter		Kemoterapi ke
		Radioterapi ke
Berobat di Rumah Sakit		No Rekam Medik
Pembiayaan Pengobatan		BPJS Kelas
Riwayat Kanker di keluarga		Hubungan Keluarga
Nama Pendamping		Hubungan dengan Pasien
Alamat Pendamping		Handphone Pendamping
Nama Pendamping 2		Hubungan dengan Pasien2
Jika tanpa pendamping menghubungi		No Handphone

Dengan ini saya menyatakan bersedia tinggal di Rumah Singgah Sasana Marsudi Husada YKI Cabang Koordinator Jawa Timur dan mematuhi segala peraturan yang berlaku.

Penerima, Pasien/ Pendamping,

Jl. Mulyorejo Indah I/8 Surabaya Phone. : 031-3892800, 92390878 Fax. : 031-3892800
E-mail : ykicabjatim@yahoo.com www.ykicabjatim.org

Gambar 1. Formulir Data Pasien

Pasien dan Pendamping

- Setiap Pasien dapat memilih jenis kamar, termasuk juga dengan Pendamping
- Setiap Pendamping bertanggung jawab untuk mendampingi pasien, dan saat Pasien tersebut mengunjungi YKI maka pasien tersebut wajib didampingi oleh satu Pendamping
- Hubungan keluarga pendamping dicatat. Sebagai contoh, suami/istri/ayah/anak/kerabat/tetangga/suster.
- Formulir yang memuat data pasien dan pendamping terdapat pada Gambar 1.

Kamar

- Kamar di YKI Jawa Timur ada berbagai tipe kamar. Saat ini terdapat 4 tipe kamar.
- Untuk memilih tipe kamar di YKI Jawa Timur, pasien menghubungi staf yang bersangkutan.
- Setiap kamar memiliki harga kamar sesuai dengan tipe kamarnya.
- Setiap kamar memiliki nomor kamar berbeda.
- Umumnya, satu kamar diisi oleh dua orang: satu pasien dan satu pendamping. Akan tetapi dalam beberapa kasus bisa saja pendampingnya lebih dari satu, tergantung tingkat keparahan pasien.

Pekerja: Pengurus dan Volunteer

- YKI memiliki sistem pekerja mereka sendiri, dan dalam pekerja tersebut sudah termasuk Pengurus dan Volunteer.
- Baik data Pengurus maupun Volunteer semua tercatat dalam database pekerja, seperti NIK, Nama, No. HP, dan Alamat.
- Namun pada Volunteer, terdapat data seperti Motivasi, Jenis Kelamin, serta pendidikan terakhir dari si Volunteer tersebut, di mana pada Pengurus hanya tambahan jabatan saja.
- Data pengurus sebagaimana pada Gambar 2.
- Formulir pendaftaran volunteer terdapat pada Gambar 3.



Gambar 2. Susunan Pengurus

YAYASAN KANKER INDONESIA
CABANG KOORDINATOR JAWA TIMUR

FORM PENDAFTARAN RELAWAN

Nama Lengkap : Dzikriulloh Al Abdu

Jenis Kelamin : Laki-laki

Tempat, Tanggal Lahir : Surabaya 4 Maret 2003

Alamat : Jl. Sutorejo no. 82

Email : Fincezz7@gmail.com

Pendidikan terakhir : SMA Div. Keperawatan Roltekkes

No.HP/WhatsApp : 088801551094

Alamat IG : @alkki

Alamat FB : Dzikriulloh Al Abdu

Alamat Tiktok :

Motivasi Menjadi Relawan : Tertarik membantu penderita kanker

Ingin Berkontribusi Dalam Hal : Promotif dan preventif dalam masalah kanker serta dalam upaya perawatan penderita kanker

Surabaya, 20 Agustus 2023

Dzikriulloh Al Abdu

Gambar 3. Formulir Pendaftaran Relawan (Volunteer)

Donatur

- Setiap donatur yang datang ke YKI dicatat nama dan alamatnya, serta diberikan penanda untuk keperluan berkas (seperti ID).
- Donatur yang datang memberikan donasi dapat berupa uang, barang, ataupun keduanya dan dicatat tanggal donasinya diberikan.

Inventaris

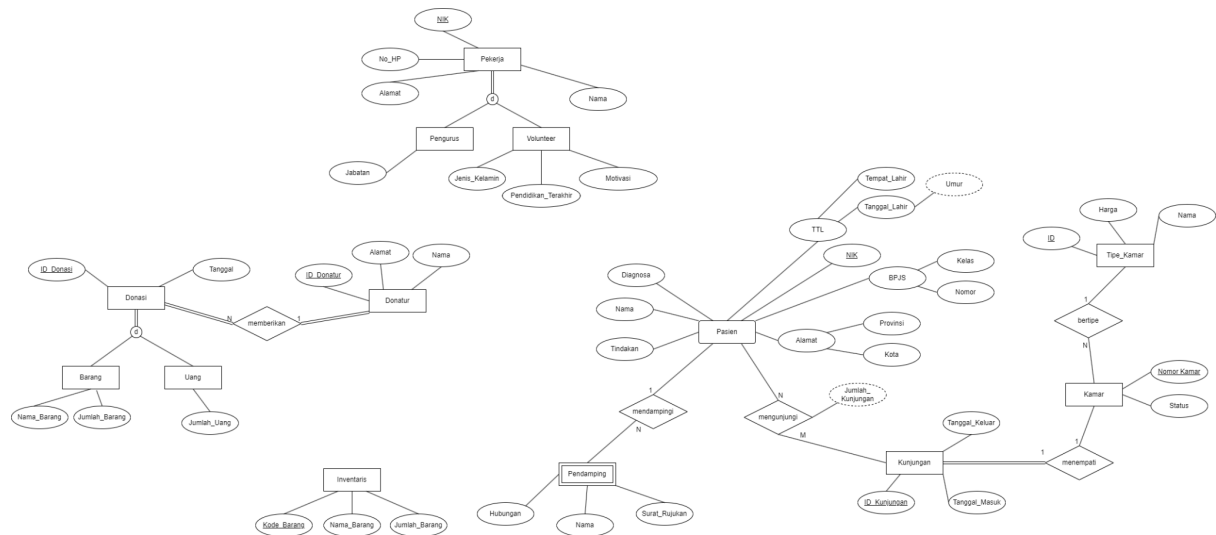
- Dalam kantor YKI Jawa Timur juga terdapat inventaris yang mencatat nama barang dan jumlah barang milik kantor untuk pendataan barang.
- Inventaris juga memberikan kode unik untuk setiap barang, dapat dilihat pada Gambar 4.

No Urut	Kode Barang	Nama Barang	Ket. Barang (Merk, Nomor Ukuran dsb)	Kuantitas
1.		Beras 25kg	Menco	2
2.		Beras 25kg	Pisang Aroma	1
3.		Gula 50kg	Nusakita	1
4.		Gula 1kg	Gulaku	9
5.		Gula 1kg	Rose Brand	4
6.		Minyak Goreng 5ltr	Ikan Dorang Mas	4
7.		Minyak Goreng 2ltr	Harumas	5
8.		Minyak Goreng 1ltr	Sovia	16
9.		Bihun Jagung	Padamu	11
10.		Mie Telor	3 Ayam	3
11.		Mie Telor	Burung Dara	3
12.		Kecap Manis	Indofood	4
13.		Detergen	Total	10
14.		Sabun Cuci Piring 5ltr	Donasi	1
15.		Sabun Cuci Lantai	Larisst	19
16.		Blender 1 set	Cosmos	1
17.		Blender 1 set	Mivako	1

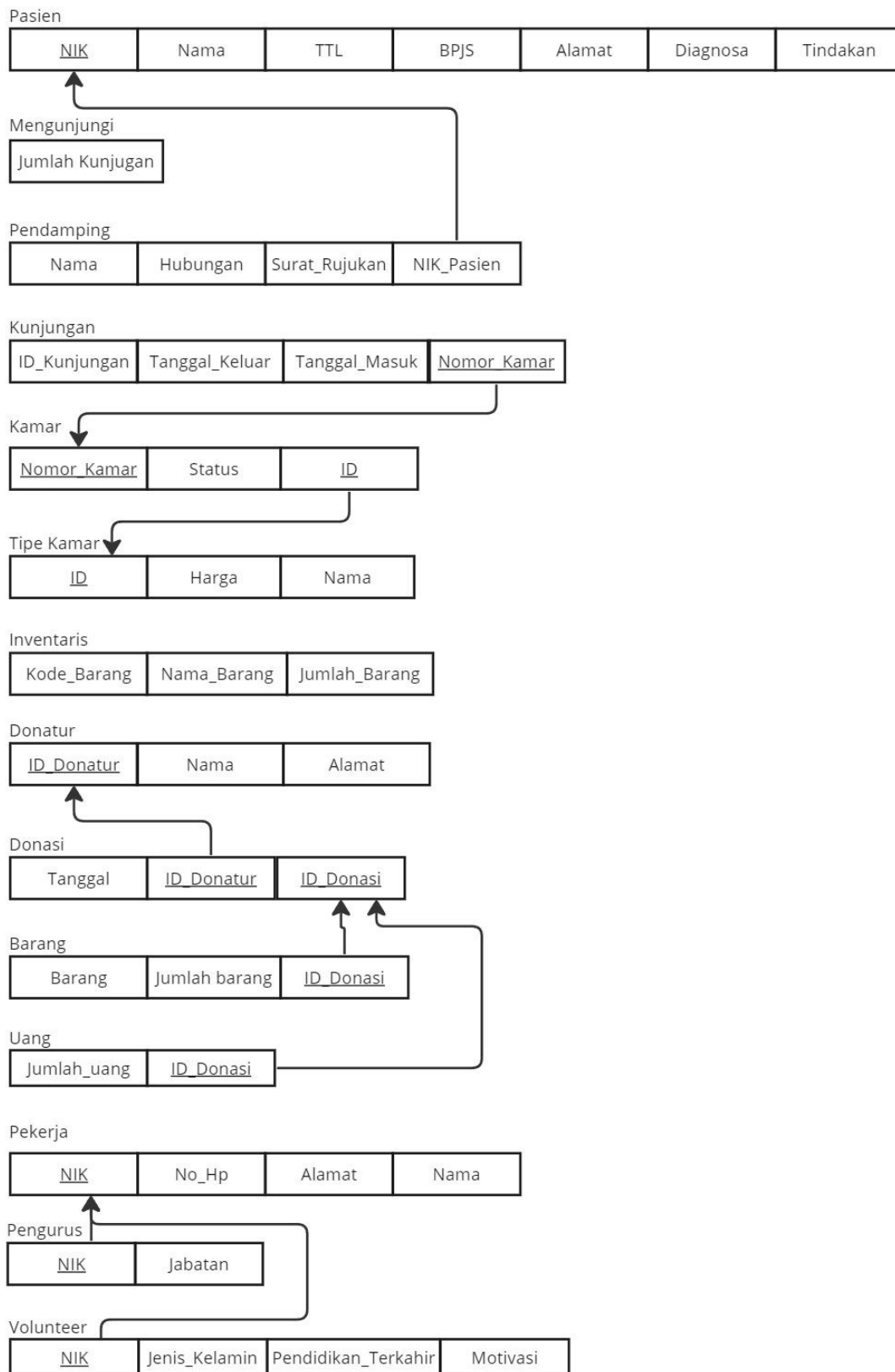
Gambar 4. Data Barang Milik YKI Jawa Timur

DATABASE DESIGN PROCESS

Enhanced Entity-Relationship (EER) Model

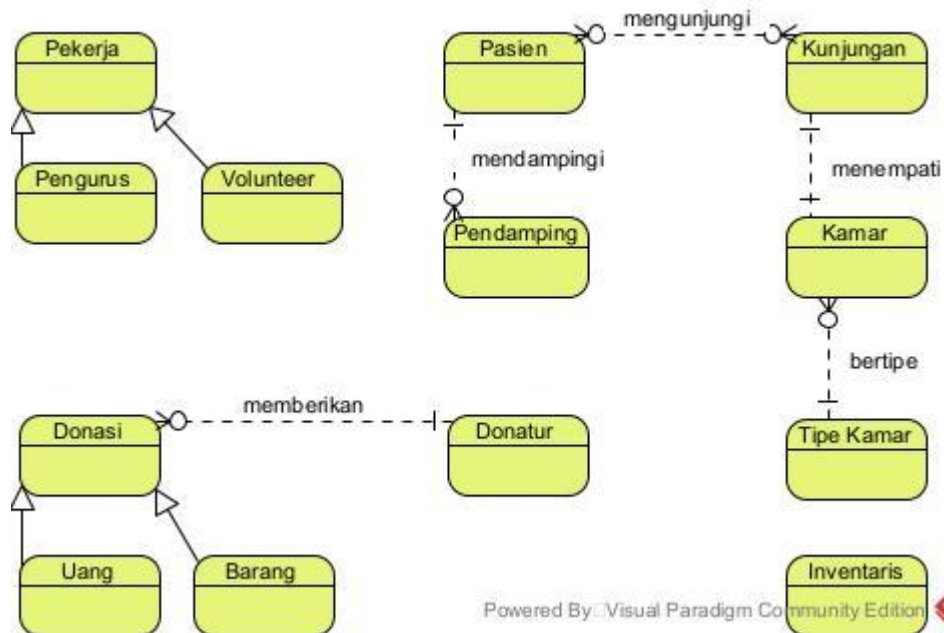


Relational Schema

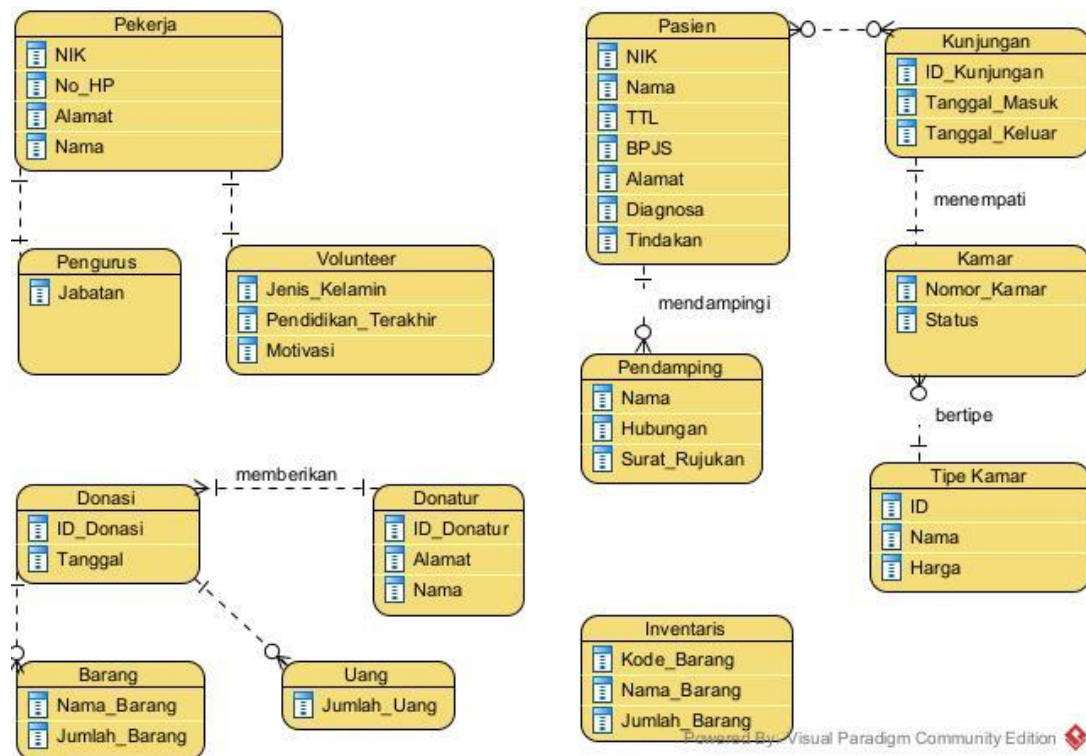


miro

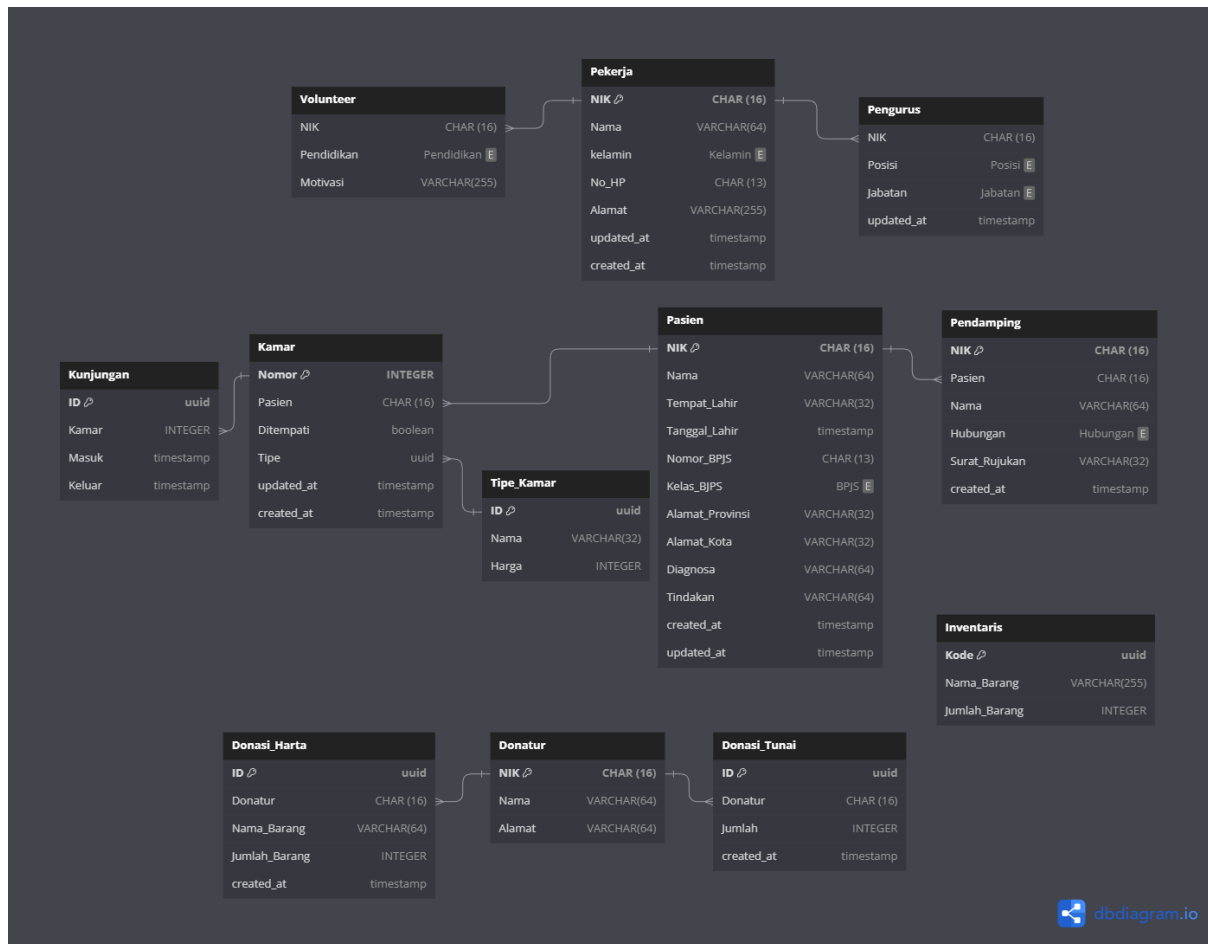
Conceptual Design



Logical Design



Physical Design



Keynote:

- Penggunaan char untuk NIK pada beberapa tabel bertujuan untuk memastikan panjang karakter sebanyak 16 karakter.
- Beberapa atribut, yaitu pendidikan, kelamin, posisi, jabatan, dan lainnya menggunakan enum karena atribut-atribut tersebut memiliki atribut yang telah ditentukan sebelumnya, seperti dijelaskan dalam bagian *Requirements*. Hal ini dilakukan untuk menghindari keambiguan jika terjadi kesalahan dan data yang salah dimasukkan ke dalam kolom.

Database Management System (DBMS)

psql (PostgreSQL) 16.0 (Debian 16.0-1.pgdg120+1)

DDL (Data Definition Language)

```
DROP TYPE IF EXISTS "Posisi" CASCADE;
CREATE TYPE "Posisi" AS ENUM (
    'Eksekutif',
    'Bendahara',
```

```
'Sekretaris',
'Organisasi',
'Pelayanan Sosial',
'Pendidikan dan Penyuluhan',
'Ilmiah',
'Umum'
);

DROP TYPE IF EXISTS "Hubungan" CASCADE;
CREATE TYPE "Hubungan" AS ENUM (
    'Pasutri',
    'Teman',
    'Saudara',
    'Orang tua',
    'Lain-Lain'
);

DROP TYPE IF EXISTS "Jabatan" CASCADE;
CREATE TYPE "Jabatan" AS ENUM (
    'Ketua',
    'Wakil',
    'Anggota'
);

DROP TYPE IF EXISTS "BPJS" CASCADE;
CREATE TYPE "BPJS" AS ENUM (
    '1',
    '2',
    '3'
);

DROP TYPE IF EXISTS "Pendidikan" CASCADE;
CREATE TYPE "Pendidikan" AS ENUM (
    'Doktor',
    'Magister',
    'Sarjana',
    'Diploma',
    'SMA',
    'SMK',
    'SMP',
    'SD'
```

```
);

DROP TYPE IF EXISTS "Kelamin" CASCADE;
CREATE TYPE "Kelamin" AS ENUM (
    'Pria',
    'Wanita'
);

DROP TABLE IF EXISTS "Pekerja" CASCADE;
CREATE TABLE "Pekerja" (
    "NIK" CHAR(16) PRIMARY KEY,
    "Nama" VARCHAR(64),
    "kelamin" "Kelamin",
    "No_HP" CHAR(13),
    "Alamat" VARCHAR(255),
    "updated_at" timestamp,
    "created_at" timestamp default current_timestamp
);

DROP TABLE IF EXISTS "Pengurus" CASCADE;
CREATE TABLE "Pengurus" (
    "NIK" CHAR(16) UNIQUE,
    "Posisi" "Posisi",
    "Jabatan" "Jabatan",
    "updated_at" timestamp
);

DROP TABLE IF EXISTS "Volunteer" CASCADE;
CREATE TABLE "Volunteer" (
    "NIK" CHAR(16) UNIQUE,
    "Pendidikan" "Pendidikan",
    "Motivasi" VARCHAR(255)
);

DROP TABLE IF EXISTS "Pasien" CASCADE;
CREATE TABLE "Pasien" (
    "NIK" CHAR(16) PRIMARY KEY,
    "Nama" VARCHAR(64),
    "Tempat_Lahir" VARCHAR(32),
    "Tanggal_Lahir" timestamp,
    "Nomor_BPJS" CHAR(13),
```




```
"Kelas_BJPS" "BPJS",
"Alamat_Provinsi" VARCHAR(32),
"Alamat_Kota" VARCHAR(32),
"Diagnosa" VARCHAR(64),
"Tindakan" VARCHAR(64),
"created_at" timestamp default current_timestamp,
"updated_at" timestamp
);

DROP TABLE IF EXISTS "Pendamping" CASCADE;
CREATE TABLE "Pendamping" (
    "NIK" CHAR(16) PRIMARY KEY,
    "Pasien" CHAR(16),
    "Nama" VARCHAR(64),
    "Hubungan" "Hubungan",
    "Surat_Rujukan" VARCHAR(32),
    "created_at" timestamp default current_timestamp
);

DROP TABLE IF EXISTS "Kunjungan" CASCADE;
CREATE TABLE "Kunjungan" (
    "ID" uuid PRIMARY KEY,
    "Kamar" INTEGER,
    "Masuk" timestamp,
    "Keluar" timestamp
);

DROP TABLE IF EXISTS "Kamar" CASCADE;
CREATE TABLE "Kamar" (
    "Nomor" INTEGER PRIMARY KEY,
    "Pasien" CHAR(16) UNIQUE,
    "Ditempati" boolean,
    "Tipe" uuid,
    "updated_at" timestamp,
    "created_at" timestamp default current_timestamp
);

DROP TABLE IF EXISTS "Tipe_Kamar" CASCADE;
CREATE TABLE "Tipe_Kamar" (
    "ID" uuid PRIMARY KEY,
    "Nama" VARCHAR(32),
```

```
"Harga" INTEGER
);

DROP TABLE IF EXISTS "Donatur" CASCADE;
CREATE TABLE "Donatur" (
    "NIK" CHAR(16) PRIMARY KEY,
    "Nama" VARCHAR(64),
    "Alamat" VARCHAR(64)
);

DROP TABLE IF EXISTS "Donasi_Tunai" CASCADE;
CREATE TABLE "Donasi_Tunai" (
    "ID" uuid UNIQUE PRIMARY KEY,
    "Donatur" CHAR(16),
    "Jumlah" INTEGER,
    "created_at" timestamp default current_timestamp
);

DROP TABLE IF EXISTS "Donasi_Harta" CASCADE;
CREATE TABLE "Donasi_Harta" (
    "ID" uuid UNIQUE PRIMARY KEY,
    "Donatur" CHAR(16),
    "Nama_Barang" VARCHAR(64),
    "Jumlah_Barang" INTEGER,
    "created_at" timestamp default current_timestamp
);

DROP TABLE IF EXISTS "Inventaris" CASCADE;
CREATE TABLE "Inventaris" (
    "Kode" uuid UNIQUE PRIMARY KEY,
    "Nama_Barang" VARCHAR(255),
    "Jumlah_Barang" INTEGER
);

ALTER TABLE
    "Pengurus"
ADD
    FOREIGN KEY ("NIK") REFERENCES "Pekerja" ("NIK");

ALTER TABLE
    "Volunteer"
```



```
ADD
    FOREIGN KEY ("NIK") REFERENCES "Pekerja" ("NIK");

ALTER TABLE
    "Pendamping"
ADD
    FOREIGN KEY ("Pasien") REFERENCES "Pasien" ("NIK");

ALTER TABLE
    "Kamar"
ADD
    FOREIGN KEY ("Pasien") REFERENCES "Pasien" ("NIK") ON UPDATE CASCADE;

ALTER TABLE
    "Kunjungan"
ADD
    FOREIGN KEY ("Kamar") REFERENCES "Kamar" ("Nomor");

ALTER TABLE
    "Kamar"
ADD
    FOREIGN KEY ("Tipe") REFERENCES "Tipe_Kamar" ("ID");

ALTER TABLE
    "Donasi_Harta"
ADD
    FOREIGN KEY ("Donatur") REFERENCES "Donatur" ("NIK");

ALTER TABLE
    "Donasi_Tunai"
ADD
    FOREIGN KEY ("Donatur") REFERENCES "Donatur" ("NIK");

CREATE OR REPLACE FUNCTION
    update_updated_at()
    RETURNS TRIGGER AS $$
BEGIN
    NEW.updated_at = now();
    RETURN NEW;
END;
$$ language 'plpgsql';
```



```
CREATE TRIGGER TRG_update_updated_at BEFORE
UPDATE
  ON "Pekerja" FOR EACH ROW EXECUTE PROCEDURE update_updated_at();

CREATE TRIGGER TRG_update_updated_at BEFORE
UPDATE
  ON "Pengurus" FOR EACH ROW EXECUTE PROCEDURE update_updated_at();

CREATE TRIGGER TRG_update_updated_at BEFORE
UPDATE
  ON "Volunteer" FOR EACH ROW EXECUTE PROCEDURE update_updated_at();

CREATE TRIGGER TRG_update_updated_at BEFORE
UPDATE
  ON "Pasien" FOR EACH ROW EXECUTE PROCEDURE update_updated_at();

CREATE TRIGGER TRG_update_updated_at BEFORE
UPDATE
  ON "Kamar" FOR EACH ROW EXECUTE PROCEDURE update_updated_at();

CREATE TRIGGER TRG_update_updated_at BEFORE
UPDATE
  ON "Donasi_Tunai" FOR EACH ROW EXECUTE PROCEDURE update_updated_at();

CREATE TRIGGER TRG_update_updated_at BEFORE
UPDATE
  ON "Donasi_Harta" FOR EACH ROW EXECUTE PROCEDURE update_updated_at();

CREATE OR REPLACE FUNCTION
  update_pasien_on_kamar_update()
  RETURNS TRIGGER AS $$
BEGIN
  IF NEW."Ditempati" = false THEN
    UPDATE
      "Kamar"
    SET
      "Pasien" = NULL
    WHERE
      "Nomor" = OLD."Nomor";
  END IF;
```

```
        RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
CREATE TRIGGER TRG_update_pasien_on_kamar_update  
AFTER  
UPDATE  
    OF "Ditempati" ON "Kamar" FOR EACH ROW EXECUTE FUNCTION  
update_pasien_on_kamar_update();
```

DML(Data Manipulation Language)

```
-- Insert dummy data for "Pekerja" table  
INSERT INTO "Pekerja" ("NIK", "Nama", "kelamin", "No_HP", "Alamat",  
"updated_at")  
VALUES  
    ('1234567890123456', 'John Doe', 'Pria', '081234567890', 'Jl. Example  
123', now()),  
    ('2345678901234567', 'Jane Doe', 'Wanita', '082345678901', 'Jl.  
Sample 456', now()),  
    ('3456789012345678', 'Bob Johnson', 'Pria', '083456789012', 'Jl. Test  
789', now()),  
    ('4567890123456789', 'Alice Smith', 'Wanita', '084567890123', 'Jl.  
Dummy 456', now()),  
    ('5678901234567890', 'Charlie Brown', 'Pria', '085678901234', 'Jl.  
Data 789', now()),  
    ('6789012345678901', 'Diana Miller', 'Wanita', '086789012345', 'Jl.  
Sample 012', now()),  
    ('7890123456789012', 'Eva Johnson', 'Wanita', '087890123456', 'Jl.  
Test 345', now()),  
    ('8901234567890123', 'Frank Miller', 'Pria', '088901234567', 'Jl.  
Example 678', now());  
  
-- Insert dummy data for "Pengurus" table  
INSERT INTO "Pengurus" ("NIK", "Posisi", "Jabatan", "updated_at")  
VALUES  
    ('1234567890123456', 'Eksekutif', 'Ketua', now()),  
    ('2345678901234567', 'Organisasi', 'Anggota', now()),  
    ('3456789012345678', 'Bendahara', 'Wakil', now()),  
    ('4567890123456789', 'Sekretaris', 'Anggota', now()),  
    ('5678901234567890', 'Pelayanan Sosial', 'Ketua', now()),  
    ('6789012345678901', 'Pendidikan dan Penyuluhan', 'Wakil', now()),
```

```
('7890123456789012', 'Ilmiah', 'Anggota', now()),
('8901234567890123', 'Umum', 'Ketua', now());

-- Insert dummy data for "Volunteer" table
INSERT INTO "Volunteer" ("NIK", "Pendidikan", "Motivasi")
VALUES
  ('1234567890123456', 'Sarjana', 'I want to contribute to the
community'),
  ('2345678901234567', 'Diploma', 'I love volunteering'),
  ('3456789012345678', 'Magister', 'Passionate about helping others'),
  ('4567890123456789', 'SMA', 'Excited to make a difference'),
  ('5678901234567890', 'Doktor', 'Committed to community service'),
  ('6789012345678901', 'Sarjana', 'Eager to contribute my skills'),
  ('7890123456789012', 'SMK', 'Inspired to give back'),
  ('8901234567890123', 'Diploma', 'Dedicated to making a positive
impact');

-- Insert dummy data for "Pasien" table
INSERT INTO "Pasien" ("NIK", "Nama", "Tempat_Lahir", "Tanggal_Lahir",
"Nomor_BPJS", "Kelas_BJPS", "Alamat_Provinsi", "Alamat_Kota",
"Diagnosa", "Tindakan", "updated_at")
VALUES
  ('3456789012345678', 'Alice Smith', 'Jakarta', '1990-01-15',
'1234567890123', '2', 'DKI Jakarta', 'Jakarta Barat', 'Fever', 'Rest',
now()),
  ('4567890123456789', 'Bob Johnson', 'Surabaya', '1985-05-20',
'9876543210987', '1', 'Jawa Timur', 'Surabaya', 'Headache',
'Medication', now()),
  ('5678901234567890', 'Charlie Brown', 'Bandung', '1998-08-30',
'5432109876543', '3', 'Jawa Barat', 'Bandung', 'Cough', 'Fluids',
now()),
  ('6789012345678901', 'Diana Miller', 'Medan', '1982-03-10',
'8765432109876', '2', 'Sumatera Utara', 'Medan', 'Fatigue', 'Rest',
now()),
  ('7890123456789012', 'Eva Johnson', 'Yogyakarta', '1995-12-25',
'3210987654321', '1', 'DI Yogyakarta', 'Yogyakarta', 'Sore throat',
'Medication', now()),
  ('8901234567890123', 'Frank Miller', 'Makassar', '1980-07-05',
'6543210987654', '3', 'Sulawesi Selatan', 'Makassar', 'Muscle pain',
'Rest', now());
```



```
-- Insert dummy data for "Pendamping" table
INSERT INTO "Pendamping" ("NIK", "Pasien", "Nama", "Hubungan",
"Surat_Rujukan", "created_at")
VALUES
    ('1111222233334444', '3456789012345678', 'David Brown', 'Saudara',
'SR123', now()),
    ('2222333344445555', '4567890123456789', 'Emma Johnson', 'Teman',
'SR456', now()),
    ('3333444455556666', '5678901234567890', 'George Smith', 'Orang tua',
'SR789', now()),
    ('4444555566667777', '6789012345678901', 'Hannah Miller',
'Lain-Lain', 'SRABC', now()),
    ('5555666677778888', '7890123456789012', 'Isaac Miller', 'Saudara',
'SRXYZ', now()),
    ('6666777788889999', '8901234567890123', 'Julia Brown', 'Teman',
'SR123ABC', now());

-- Insert dummy data for "Tipe_Kamar" table
INSERT INTO "Tipe_Kamar" ("ID", "Nama", "Harga")
VALUES
    ('123e4567-e89b-12d3-a456-426614174003', 'Standard', 500000),
    ('234e5678-e89b-12d3-a456-426614174004', 'Deluxe', 1000000),
    ('345e6789-e89b-12d3-a456-426614174005', 'Suite', 1500000);

-- Insert dummy data for "Kamar" table
INSERT INTO "Kamar" ("Nomor", "Pasien", "Ditempati", "Tipe",
"updated_at")
VALUES
    (1, '3456789012345678', true, '123e4567-e89b-12d3-a456-426614174003',
now()),
    (2, '4567890123456789', true, '234e5678-e89b-12d3-a456-426614174004',
now()),
    (3, '5678901234567890', true, '345e6789-e89b-12d3-a456-426614174005',
now()),
    (4, '6789012345678901', true, '123e4567-e89b-12d3-a456-426614174003',
now()),
    (5, '7890123456789012', true, '234e5678-e89b-12d3-a456-426614174004',
now()),
    (6, '8901234567890123', true, '345e6789-e89b-12d3-a456-426614174005',
now());
```



```
-- Insert dummy data for "Kunjungan" table
INSERT INTO "Kunjungan" ("ID", "Kamar", "Masuk", "Keluar")
VALUES
  ('123e4567-e89b-12d3-a456-426614174001', 1, '2023-01-01',
'2023-01-03'),
  ('234e5678-e89b-12d3-a456-426614174002', 2, '2023-01-02',
'2023-01-04'),
  ('345e6789-e89b-12d3-a456-426614174003', 3, '2023-01-03',
'2023-01-05'),
  ('456e7890-e89b-12d3-a456-426614174004', 1, '2023-01-04',
'2023-01-06'),
  ('567e8901-e89b-12d3-a456-426614174005', 2, '2023-01-05',
'2023-01-07'),
  ('678e9012-e89b-12d3-a456-426614174006', 3, '2023-01-06',
'2023-01-08'),
  ('789e0123-e89b-12d3-a456-426614174007', 1, '2023-01-07',
'2023-01-09'),
  ('890e1234-e89b-12d3-a456-426614174008', 2, '2023-01-08',
'2023-01-10');

-- Insert dummy data for "Donatur" table
INSERT INTO "Donatur" ("NIK", "Alamat", "Nama")
VALUES
  ('1111222233334444', 'Jl. Donator 789', 'Karen Smith'),
  ('2222333344445555', 'Jl. Generous 456', 'Larry Johnson'),
  ('3333444455556666', 'Jl. Kindhearted 123', 'Megan Brown'),
  ('4444555566667777', 'Jl. Benevolent 987', 'Oscar Miller'),
  ('5555666677778888', 'Jl. Philanthropist 654', 'Penelope Williams'),
  ('6666777788889999', 'Jl. Charitable 321', 'Quincy Davis'),
  ('7777888899990000', 'Jl. Giving 567', 'Rita Taylor'),
  ('8888999900001111', 'Jl. Compassionate 890', 'Steve Wilson');

-- Insert dummy data for "Donasi_Tunai" table
INSERT INTO "Donasi_Tunai" ("ID", "Donatur", "Jumlah", "created_at")
VALUES
  ('345e6789-e89b-12d3-a456-426614174005', '1111222233334444', 1000000,
now()),
  ('456e7890-e89b-12d3-a456-426614174006', '2222333344445555', 500000,
now()),
  ('567e8901-e89b-12d3-a456-426614174007', '3333444455556666', 1500000,
now()),
```



```
( '678e9012-e89b-12d3-a456-426614174008', '4444555566667777', 2000000,
now()),
( '789e0123-e89b-12d3-a456-426614174009', '5555666677778888', 800000,
now()),
( '890e1234-e89b-12d3-a456-426614174010', '6666777788889999', 1200000,
now()),
( '901e2345-e89b-12d3-a456-426614174011', '7777888899990000', 600000,
now()),
( '012e3456-e89b-12d3-a456-426614174012', '8888999900001111', 900000,
now());

-- Insert dummy data for "Donasi_Harta" table
INSERT INTO "Donasi_Harta" ("ID", "Donatur", "Nama_Barang",
"Jumlah_Barang", "created_at")
VALUES
( '123e4567-e89b-12d3-a456-426614174013', '1111222233334444',
'Clothes', 50, now()),
( '234e5678-e89b-12d3-a456-426614174014', '2222333344445555', 'Books',
100, now()),
( '345e6789-e89b-12d3-a456-426614174015', '3333444455556666', 'Toys',
30, now()),
( '456e7890-e89b-12d3-a456-426614174016', '4444555566667777',
'Electronics', 20, now()),
( '567e8901-e89b-12d3-a456-426614174017', '5555666677778888',
'Furniture', 10, now()),
( '678e9012-e89b-12d3-a456-426614174018', '6666777788889999',
'Appliances', 15, now()),
( '789e0123-e89b-12d3-a456-426614174019', '7777888899990000',
'Jewelry', 5, now()),
( '890e1234-e89b-12d3-a456-426614174020', '8888999900001111',
'Artwork', 8, now());

-- ON UPDATE TRIGGER TESTING
UPDATE "Kamar"
SET "Ditempati" = false
WHERE "Nomor" IN (4, 5, 6);
```

DQL (Data Query Language)

```
-- Retrieve all workers with their positions:
SELECT
    "Nama",
```

```
"Posisi"
FROM
  "Pekerja"
  JOIN "Pengurus" ON "Pekerja"."NIK" = "Pengurus"."NIK";

-- Retrieve all volunteers and their motivations:
SELECT
  "Nama",
  "Motivasi"
FROM
  "Volunteer"
  JOIN "Pekerja" ON "Volunteer"."NIK" = "Pekerja"."NIK";

-- Retrieve information about patients and their diagnoses:
SELECT
  "Nama",
  "Diagnosa",
  "Tindakan"
FROM
  "Pasien";

-- Retrieve the list of donors and their donation amounts:
SELECT
  "Nama",
  "Jumlah"
FROM
  "Donatur"
  JOIN "Donasi_Tunai" ON "Donatur"."NIK" = "Donasi_Tunai"."Donatur";

-- Retrieve the list of rooms with their occupancy status and type:
SELECT
  "Nomor",
  "Ditempati",
  "Nama" as "Tipe_Kamar"
FROM
  "Kamar"
  JOIN "Tipe_Kamar" ON "Kamar"."Tipe" = "Tipe_Kamar"."ID";

-- Retrieve the list of visitors and their relationships with patients:
SELECT
  "Nama",
```



```
"Hubungan"
FROM
    "Pendamping";

-- Retrieve the details of cash donations with donor information:
SELECT
    "Donatur"."Nama" as "Donor_Name",
    "Jumlah",
    "created_at"
FROM
    "Donasi_Tunai"
JOIN "Donatur" ON "Donasi_Tunai"."Donatur" = "Donatur"."NIK";

-- Retrieve the details of inpatient visits, including room
information:
SELECT
    "Kunjungan"."ID",
    "Pasien"."Nama" as "Patient_Name",
    "Masuk",
    "Keluar",
    "Kamar"."Nomor" as "Room_Number",
    "Tipe_Kamar"."Nama" as "Room_Type"
FROM
    "Kunjungan"
JOIN "Kamar" ON "Kunjungan"."Kamar" = "Kamar"."Nomor"
JOIN "Tipe_Kamar" ON "Kamar"."Tipe" = "Tipe_Kamar"."ID"
JOIN "Pasien" ON "Kamar"."Pasien" = "Pasien"."NIK";
```